Differential Abundance methods- Simulation Design

Saritha Kodikara

February 18, 2022

Contents

1 Simulation Design

To assess the performance of the differential abundance methods discussed in this review paper, we simulated longitudinal count data from a generalized linear model with a negative binomial distribution using the 'tscount' R package (Liboschik, Fokianos, and Fried 2017). We estimated realistic parameter values for dispersion and Autoregressive (AR) coefficient to be used in the simulation using the pregnancy data (DiGiulio et al. 2015). Based on these estimated values nine case scenarios were created with three dispersion (i.e., noise) values [0.1;0.3;0.6] and three AR values [0.04; 0.2; 0.4].

%

```
#Simulation function
simulate<-function(mod,para, nIndiv, nTime,disper, nTaxa, nSc1, nSc2, nSc3, meta){

  TAXA<-setNames(data.frame(matrix(ncol = nTaxa,
                nrow = nIndiv*nTime)),paste0("Taxa_",1:nTaxa))

  for(i in 1:nTaxa){
    if(i<=nSc1){#Time
      k=1
    }else if(i<=nSc1+nSc2){#Group
      k=2
    }else if(i<=nSc1+nSc2+nSc3){#Time+Group+Time*Group
      k=3
    }else {#No
      k=4
    }
    count<-c()
    for(j in 1:nIndiv){
      t<-c(tsglm.sim(n=nTime, param = para[[k]], model=model,
                     xreg=matrix(c(1:nTime,
                     rep(meta$Group[meta$Indiv==j][1],nTime),
                     1:nTime*rep(meta$Group[meta$Indiv==j][1],nTime)),ncol=3),
                     link="identity",
                     distr="nbinom", distrcoefs=c(size=1/disper))$ts)
      count<-c(count,t)
```

Table 1: Parameter values used in scenario 1 for 300 taxa.

Effect - Number of taxa	**Intercept**	**Time**	**Group**	**Group*Time**
Time - 10	Uniform(0,5)	1.5	0	0
Group - 10	Uniform(0,5)	0	13	0
Group+Time+Group*Time -10	Uniform(0,5)	1.5	13	5
No - 270	Uniform(0,5)	0	0	0

```r
    }
    while(!any(count==0)){#At least one zero (Needed to run ZIGMM)
      count<-c()
      for(j in 1:nIndiv){
        t<-c(tsglm.sim(n=nTime, param = para[[k]], model=model,
                       xreg=matrix(c(1:nTime,rep(meta$Group[meta$Indiv==j][1],nTime),
                            1:nTime*rep(meta$Group[meta$Indiv==j][1],nTime)),ncol=3),
                       link="identity",
                       distr="nbinom", distrcoefs=c(size=1/disper))$ts)
        count<-c(count,t)
      }
    }
    TAXA[,i]<-count
  }


  return(TAXA)
}
```

1.1 Data Simulation- Scenario 1

In this section, we simulate data under Scenario 1 (i.e., dispersion=0.1; AR value=0.04). For Scenario 1, 50 data sets were simulated with 300 profiles (i.e. taxa) measured on 10-time points and twenty individuals belonging to two groups. These 300 profiles consisted of five profiles that were differentially expressed trough time only, five profiles that were differentially expressed trough group only, five profiles that were differentially expressed trough time and group and 270 profiles that were without any differentially expressed effect (i.e., noise profiles) (see Table 1). All these profiles had at-least one zero count between time 0 and time 10 for at-least one individual. This condition was included into the simulation as to avoid errors in ZIGMM and FZINBMM model fitting.

```r
set.seed(1234)
model <- list(past_obs=1) #Only 1 AR parameter
disp_1<-0.1
param_1 <- list(list(intercept=runif(1,0,5), past_obs=0.04,  xreg=c(1.5,0,0)),
                list(intercept=runif(1,0,5), past_obs=0.04,  xreg=c(0,13,0)),
                list(intercept=runif(1,0,5), past_obs=0.04,  xreg=c(1.5,13,5)),
                list(intercept=runif(1,0,5), past_obs=0.04,  xreg=c(0,0,0)))

nIndiv=20;nTime=10
metaDF<-data.frame(Time=rep(c(1:nTime),nIndiv),
                   Indiv=rep(1:nIndiv,each=nTime),
                   Group=rep(0:1,each=nIndiv*nTime/2))

n_iter <- 50 # Number of iterations of the loop
list_of_frames <- replicate(n_iter, data.frame())
```

```r
# Initializes the progress bar
pb <- txtProgressBar(min = 0,      # Minimum value of the progress bar
                     max = n_iter, # Maximum value of the progress bar
                     style = 3,    # Progress bar style
                     width = 50,   # Progress bar width
                     char = "=")   # Character used to create the bar
```

```
##   |                                                          |
```

```r
for(i in 1:n_iter) {

  #---------------------
  # Code to be executed
  #---------------------
  df_Scenario1<-simulate(model,param_1,nIndiv,nTime, disp_1, 300, 10,10,10,metaDF)
  list_of_frames[[i]] <-df_Scenario1
  #---------------------

  # Sets the progress bar to the current state
  setTxtProgressBar(pb, i)
}
```

```
##   |                                                          |=
```

```r
close(pb) # Close the connection
```

```r
c_Sc1<-lapply(list_of_frames, function(x) {
  mutate(x,Library_size=rowSums(x))
})

ra_Sc1<-lapply(c_Sc1, function(x) {
  x[,-301]/x$Library_size
})

saveRDS(metaDF,file = "Data/df_meta.Rdata")
saveRDS(c_Sc1,file = "Data/count_Scenario1.Rdata")
saveRDS(ra_Sc1,file = "Data/RA_Scenario1.Rdata")
```

```r
dfcount<- cbind(metaDF,c_Sc1[[1]])
dfRA<- cbind(metaDF,ra_Sc1[[1]])


p1<-dfcount %>%
  ggplot(aes(x=Time, y=Taxa_1,  colour= as.factor(Group),
             group =  as.factor(Indiv), shape= as.factor(Group),
             linetype = as.factor(Group)))+
  geom_line()+ geom_point()+ggtitle("a")+ylab("Count")+
  labs( linetype = "Group", shape = "Group", color="Group") +
  scale_color_manual(values=c("steelblue","darkviolet"))+
  theme(legend.position = "none")

p11<-dfcount %>%
  ggplot(aes(x=Time, y=Taxa_11,  colour= as.factor(Group),
             group =  as.factor(Indiv), shape= as.factor(Group),
             linetype = as.factor(Group)))+
  geom_line()+ geom_point()+ggtitle("b")+ylab("Count")+
  labs( linetype = "Group", shape = "Group", color="Group")  +
  scale_color_manual(values=c("steelblue","darkviolet"))+
  theme(legend.position = "none")

p21<-dfcount %>%
  ggplot(aes(x=Time, y=Taxa_21,  colour= as.factor(Group),
             group =  as.factor(Indiv), shape= as.factor(Group),
             linetype = as.factor(Group)))+
  geom_line()+ geom_point()+ggtitle("c")+ylab("Count")+
  labs( linetype = "Group", shape = "Group", color="Group")  +
  scale_color_manual(values=c("steelblue","darkviolet"))+
  theme(legend.position = "none")

p31<-dfcount %>%
  ggplot(aes(x=Time, y=Taxa_31,  colour= as.factor(Group),
             group =  as.factor(Indiv), shape= as.factor(Group),
             linetype = as.factor(Group)))+
  geom_line()+ geom_point()+ggtitle("d")+ylab("Count")+
  labs( linetype = "Group", shape = "Group", color="Group")  +
  scale_color_manual(values=c("steelblue","darkviolet"))+
  theme(legend.position = "none")


plot1<-p1|p11|p21|p31

p1.1<-dfRA %>%
  ggplot(aes(x=Time, y=Taxa_1,   colour= as.factor(Group),
             group =  as.factor(Indiv), shape= as.factor(Group),
             linetype = as.factor(Group)))+
  geom_line()+ geom_point()+ggtitle("A")+ylab("Relative Abundance")+
  labs( linetype = "Group", shape = "Group", color="Group")  +
  scale_color_manual(values=c("steelblue","darkviolet"))+
  theme(legend.position = "none")


p11.1<-dfRA %>%
  ggplot(aes(x=Time, y=Taxa_11,  colour= as.factor(Group),
             group =  as.factor(Indiv), shape= as.factor(Group),
             linetype = as.factor(Group)))+
  geom_line()+ geom_point()+ggtitle("B")+ylab("Relative Abundance")+
  labs( linetype = "Group", shape = "Group", color="Group")  +
  scale_color_manual(values=c("steelblue","darkviolet"))+
  theme(legend.position = "none")

p21.1<-dfRA %>%
  ggplot(aes(x=Time, y=Taxa_21,   colour= as.factor(Group),
             group =  as.factor(Indiv), shape= as.factor(Group),
             linetype = as.factor(Group)))+
  geom_line()+ geom_point()+ggtitle("C")+ylab("Relative Abundance")+
  labs( linetype = "Group", shape = "Group", color="Group")  +
  scale_color_manual(values=c("steelblue","darkviolet"))+
  theme(legend.position = "none")
```

```
p31.1<-dfRA %>%
  ggplot(aes(x=Time, y=Taxa_31,   colour= as.factor(Group),
        group =  as.factor(Indiv), shape= as.factor(Group),
        linetype = as.factor(Group)))+
  geom_line()+ geom_point()+ggtitle("D")+ylab("Relative Abundance")+
  labs( linetype = "Group", shape = "Group", color="Group") +
  scale_color_manual(values=c("steelblue","darkviolet"))+
  theme(legend.position = "none")


(p1|p11|p21|p31)/(p1.1|p11.1|p21.1|p31.1)
```

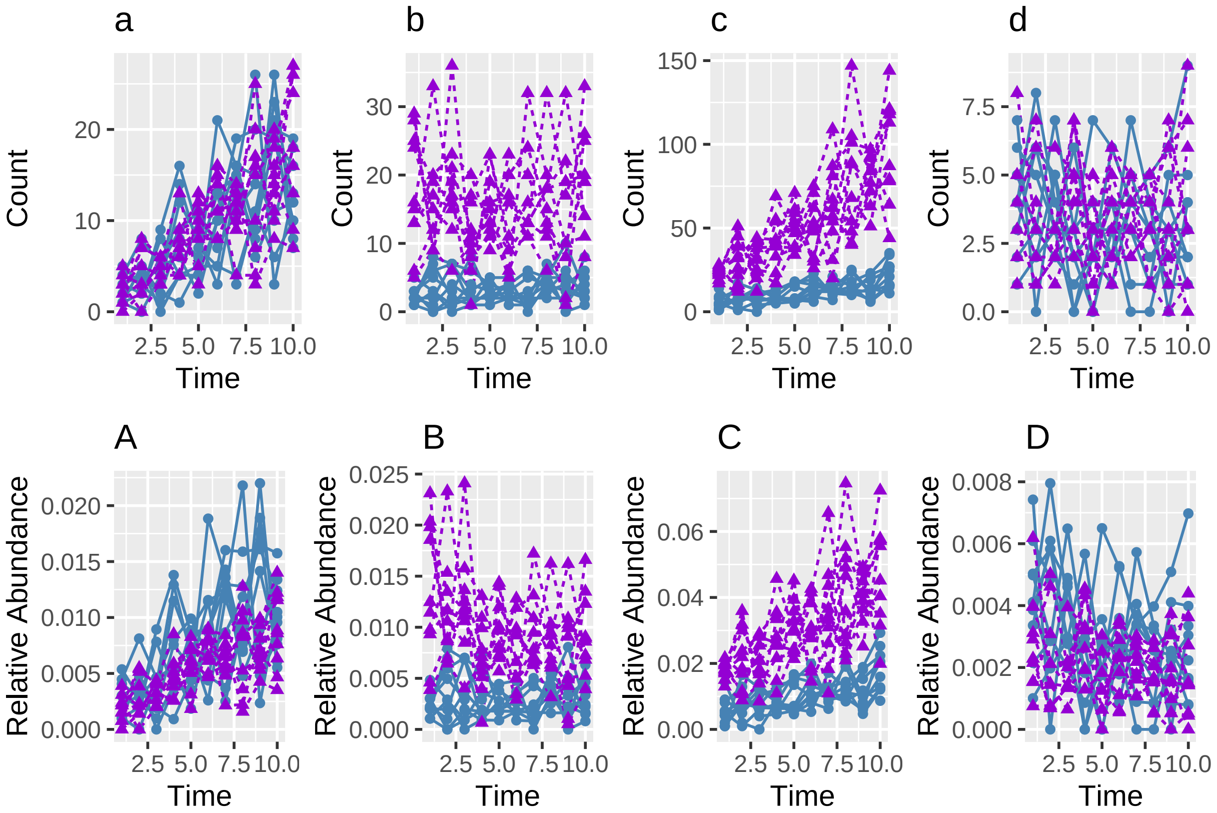

Figure 1: Figure 1, Some simulated taxa profiles with time (A,a); group (B,b); group * time (C,c); and no effect (D,d) with 0.1 dispersion and 0.04 AR. Top panel shows the counts, while the bottom shows the relative abundance values

References

DiGiulio, Daniel B, Benjamin J Callahan, Paul J McMurdie, Elizabeth K Costello, Deirdre J Lyell, Anna Robaczewska, Christine L Sun, et al. 2015. "Temporal and Spatial Variation of the Human Microbiota During Pregnancy." *Proceedings of the National Academy of Sciences* 112 (35): 11060–5.

Liboschik, Tobias, Konstantinos Fokianos, and Roland Fried. 2017. "Tscount: An R Package for Analysis of Count Time Series Following Generalized Linear Models." *Journal of Statistical Software* 82 (1): 1–51.